Automatica 70 (2016) 109–120

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Irredundant lattice representations of continuous piecewise affine functions☆

Jun Xu [a,b], Ton J.J. van den Boom [b], Bart De Schutter [b], Shuning Wang [c,d]

[a] Department of Automation, China University of Petroleum (Beijing), China
[b] Delft Center for Systems and Control, Delft University of Technology, The Netherlands
[c] Department of Automation, Tsinghua University, Beijing, China
[d] Tsinghua National Laboratory for Information Science and Technology (TNList), China

ARTICLE INFO

Article history:
Received 3 November 2014
Received in revised form
29 June 2015
Accepted 6 March 2016
Available online 15 April 2016

Keywords:
Piecewise affine function
Irredundant representation
Lattice representation

ABSTRACT

In this paper, we revisit the full lattice representation of continuous piecewise affine (PWA) functions and give a formal proof of its representation ability. Based on this, we derive the irredundant lattice PWA representations through removal of redundant terms and literals. Necessary and sufficient conditions for irredundancy are proposed. Besides, we explain how to remove terms and literals in order to ensure irredundancy. An algorithm is given to obtain an irredundant lattice PWA representation. In the worked examples, the irredundant lattice PWA representations are used to express the optimal solution of explicit model predictive control problems, and the results turn out to be much more compact than those given by a state-of-the-art algorithm.

1. Introduction

A continuous piecewise affine (PWA) function is a nonlinear function with affine components defined on polyhedral subregions. It is demonstrated in Wilkinson (1963) that any continuous PWA function can be expressed by a min–max or max–min composition of its affine components,

$$f=\min_{i=1,\ldots,N_1}\{\max_{j\in\bar{I}_i}\{\ell_j\}\}, \tag{1}$$

or

$$f=\max_{i=1,\ldots,N_2}\{\min_{j\in\tilde{I}_i}\{\ell_j\}\}, \tag{2}$$

in which ℓ_j is an affine function, N_1 and N_2 are integers, and $\bar{I}_i$ and $\tilde{I}_i$ are index sets. In Tarela and Martinez (1999), formal proofs are given demonstrating that any continuous PWA function can be described by (1) and (2), which are then called lattice PWA representations. They also appeared in De Schutter and van den Boom (2004) and Gunawardena (1994). We call (1) the conjunctive form and (2) disjunctive form. In Bartels, Kuntz, and Scholtes (1995), Ovchinnikov (2002) and Ovchinnikov (2010), the representation ability of (1) and (2) is also proved.

Among all these papers Bartels et al. (1995), De Schutter and van den Boom (2004), Gunawardena (1994), Ovchinnikov (2002), Ovchinnikov (2010), Tarela and Martinez (1999) and Wilkinson (1963), only Tarela and Martinez (1999) and Wilkinson (1963) give methods for determining the parameters N_1, $\bar{I}_i$ in (1) and N_2, $\tilde{I}_i$ in (2). However, Wilkinson (1963) only illustrates how to determine the parameters for a 1-dimensional example and does not provide a formal proof. Moreover, it is demonstrated in Ovchinnikov (2010) that an important assumption is not stated in the proofs in Tarela and Martinez (1999), while without that assumption the conclusions do not hold. In this paper, we mainly focus on the disjunctive lattice PWA representation (2), and give a proof concerning the representation ability as well as the determination of the parameters. The results can be easily extended to the conjunctive case due to duality.

There are also other methods for representing PWA functions (Breiman, 1993; Julián, Desages, & Agamennoni, 1999; Wang,

☆ This research was partially supported by China Scholarship Council (CSC), National Natural Science Foundation of China (61104218; 61134012; 61473165), the National Basic Research Program of China (973 program: 2012CB720505), the Science Foundation of China University of Petroleum (YJRC-2013-12). The material in this paper was partially presented at the 53rd IEEE Conference on Decision and Control, December 15–17, 2014, Los Angeles, CA, USA. This paper was recommended for publication in revised form by Associate Editor Mario Sznaier under the direction of Editor Richard Middleton.

E-mail addresses: xujunqgy@gmail.com (J. Xu), a.j.j.vandenBoom@tudelft.nl (T.J.J. van den Boom), b.deschutter@tudelft.nl (B. De Schutter), swang@mail.tsinghua.edu.cn (S. Wang).

http://dx.doi.org/10.1016/j.automatica.2016.03.018
0005-1098/

Huang, & Junaid, 2008; Wang & Sun, 2005; Xu, Huang, & Wang, 2009). The methods of Breiman (1993) can only represent continuous PWA functions in 1 dimension. The representations of Julián et al. (1999) and Xu et al. (2009) can only represent continuous PWA functions of which the domain is partitioned into simplices or the union of simplices. Although the representations proposed in Wang et al. (2008) and Wang and Sun (2005) can represent any continuous PWA function, the parameters in the expression of Wang and Sun (2005) are hard to derive and the number of parameters in the expression of Wang et al. (2008) is large. Conversely, we will show in Section 2 that the integer N_2 and the index set $\tilde{I}_i$ in (2) are not hard to derive.

Lattice PWA representations have been used to express the solution of explicit model predictive control (MPC) problems in Wen, Ma, and Ydstie (2009). In MPC, the control action is obtained by solving a finite-horizon open-loop optimal control problem at each sampling instant. At the next time step, a new optimal control problem based on new measurements of the state is solved over a shifted horizon. The optimization relies on a prediction model for predicting future outputs of the system, can take into account input and output constraints, and minimizes a performance criterion (Bemporad, Borrelli, & Morari, 2002). When the constraints are affine, a continuous PWA control law arises if the performance criterion in the optimization problem of MPC is convex quadratic or polyhedral. Then, the optimal solution can be computed offline, and the cost of online optimization can be reduced to that of online evaluation of a continuous PWA function. This is exactly what "explicit" means.

The corresponding continuous PWA optimal solution can be computed using multi-parametric linear or quadratic programming through e.g. the MPT Toolbox (Herceg, Kvasnica, Jones, & Morari, 2013) and stored as a collection of local affine functions and subregions. For online evaluation, many papers are dedicated to solving a point location problem, i.e., determining the subregion the present state is located in, and then finding the corresponding local affine function (Christophersen, Kvasnica, Jones, & Morari, 2007; Herceg, Mariéthoz, & Morari, 2013; Tøndel, Johansen, & Bemporad, 2003b). The online search complexity is logarithmic in the number of subregions (Herceg, Mariéthoz et al., 2013; Tøndel et al., 2003b) or linear in the number of subregions (Christophersen et al., 2007). For this kind of methods, the online search can be accelerated through storing additional information apart from the polyhedral partition, such as search tree and adjacency information.

On the other hand, some papers reduce the offline storage complexity by avoiding the storage of the polyhedral information (Baotic, Borrelli, Bemporad, & Morari, 2008; Jones, Grieder, & Raković, 2006). For the case of linear cost function, both methods store only the optimal value function; the online evaluation complexity for Baotic et al. (2008) is linear in the number of subregions while for the method of Jones et al. (2006) it is logarithmic. However, for the quadratic cost case, the method in Jones et al. (2006) is not applicable and the procedure of Baotic et al. (2008) has to store the information of the descriptor function as well as the ordering of local affine functions in neighboring polyhedra. Hence, it is of great value to find a method to reduce offline storage complexity for both the linear and the quadratic case.

For a continuous PWA controller derived in the linear or the quadratic case, through determining the parameters of (1), the lattice PWA function is used to represent the controller in Wen et al. (2009). For online evaluation, the current state is then directly substituted into expression (1) and the optimal solution results. By removing redundant parameters in the lattice PWA representations, both the storage requirements and the online complexity can be reduced. However, the simplification lemmas in Wen et al. (2009) have limitations and the result cannot guaranteed to be irredundant. Hence, in the current paper, we aim to give irredundant lattice PWA representations.

The paper is organized as follows. The next section introduces the full lattice PWA representation, and gives a proof of its representation ability. The *irredundant* lattice PWA representations are derived in Section 3, including necessary and sufficient conditions for irredundancy and the algorithm for obtaining an irredundant lattice PWA representation. The offline preprocessing and online evaluation complexity of the irredundant lattice PWA representations are also analyzed. In Section 4, two worked examples are given, in which the irredundant lattice PWA representations are applied to express the solutions of explicit MPC problems. Finally, the paper ends with conclusions in Section 5.

2. Full lattice PWA representation

Definition 1 (*Chua & Deng, 1988*)**.** A function $f : \mathbb{D} \to \mathbb{R}$, where $\mathbb{D} \subseteq \mathbb{R}^n$ is convex, is said to be continuous PWA if it is continuous on the domain $\mathbb{D}$ and the following conditions are satisfied:

(1) The domain space $\mathbb{D}$ can be divided into a finite number of nonempty convex polyhedra, i.e., $\mathbb{D} = \cup_{i=1}^{\hat{N}} \Omega_i$, $\Omega_i \neq \emptyset$, the polyhedra are closed and have non-overlapping interiors, $\mathrm{int}(\Omega_i) \cap \mathrm{int}(\Omega_j) = \emptyset$, $\forall i, j \in \{1, \ldots, \hat{N}\}$, $i \neq j$. These polyhedra are also called subregions. The boundaries of the polyhedra are $(n-1)$-dimensional hyperplanes.

(2) In each subregion Ω_i, f equals a local affine function $\ell_{\mathrm{loc}(i)}$,

$$f(x) = \ell_{\mathrm{loc}(i)}(x), \quad \forall x \in \Omega_i.$$

It is important to note that in Definition 1 some local affine function may appear in different subregions, i.e., $\ell_{\mathrm{loc}(i_1)} = \cdots = \ell_{\mathrm{loc}(i_s)}$ for different $i_1, \ldots, i_s \in \{1, \ldots, \hat{N}\}$. We collect all the local affine functions and select those distinct ones, labeling them as $\ell_1, \ldots, \ell_M$. So $\mathrm{loc}(i) \in \{1, \ldots, M\}$ and no two affine functions ℓ_i and ℓ_j with $i, j \in \{1, \ldots, M\}$, $i \neq j$, are identical. Therefore, there can be more subregions than distinct affine functions.

We further partition each subregion Ω_i $(i = 1, \ldots, \hat{N})$ into so called base regions $\mathbb{D}_{i,t}$ with $t = 1, \ldots, m_i$, to make sure that no other affine function intersects with $\ell_{\mathrm{loc}(i)}$ at some point in the interior of $\mathbb{D}_{i,t}$, i.e.,

$$\{x | \ell_j(x) = \ell_{\mathrm{loc}(i)}(x), j \neq \mathrm{loc}(i)\} \cap \mathrm{int}(\mathbb{D}_{i,t}) = \emptyset. \tag{3}$$

The following lemma defines the partition.

Lemma 1. *For any $i \in \{1, \ldots, \hat{N}\}$, there is a partition of the subregion Ω_i*

$$\Omega_i = \cup_{t=1}^{m_i} \mathbb{D}_{i,t} \tag{4}$$

such that the following holds,

(1) *The set $\mathrm{int}(\mathbb{D}_{i,t})$ is nonempty.*

(2) *For each $\mathbb{D}_{i,t}$, we have*

$$I_{\geq,i,t} \cup I_{\leq,i,t} = \{1, \ldots, M\}, \tag{5}$$

in which $I_{\geq,i,t} = \{j | \ell_j(x) \geq \ell_{\mathrm{loc}(i)}(x), \ \forall x \in \mathbb{D}_{i,t}\}$ and $I_{\leq,i,t} = \{j | \ell_j(x) \leq \ell_{\mathrm{loc}(i)}(x), \ \forall x \in \mathbb{D}_{i,t}\}$.

(3) *For all $i, j \in \{1, \ldots, \hat{N}\}$, $\bar{t} \in \{1, \ldots, m_i\}$, $\hat{t} \in \{1, \ldots, m_j\}$, $\bar{t} \neq \hat{t}$ or $i \neq j$, the following holds,*

$$\mathrm{int}(\mathbb{D}_{i,\bar{t}}) \cap \mathrm{int}(\mathbb{D}_{j,\hat{t}}) = \emptyset. \tag{6}$$

The proof of Lemma 1 as well as the time complexity of the partition process is given in Appendix A.